

Deep Learning Course - Analisis Klasifikasi Dataset (CNN Model)

Muhammad Naufal Firdaus_1103220083

1. **Arsitektur CNN dengan X lapisan konvolusi menghasilkan akurasi training 98% tetapi akurasi validasi 62%. Jelaskan fenomena vanishing gradient yang mungkin terjadi pada lapisan awal, dan bagaimana cara memitigasinya! Mengapa penambahan Batch Normalization setelah lapisan konvolusi ke-Y justru memperburuk generalisasi, serta strategi alternatif untuk menstabilkan pembelajaran?**

Fenomena Vanishing Gradient:

Ketika CNN terlalu dalam (banyak lapisan konvolusi), gradien pada lapisan awal bisa menjadi sangat kecil selama backpropagation, sehingga bobot pada lapisan tersebut hampir tidak diperbarui. Hal ini menyebabkan model sulit belajar fitur low-level yang penting dari input.

Penyebab:

- Aktivasi seperti sigmoid atau tanh menyebabkan gradien mengecil saat disusun dalam banyak lapisan.
- Inisialisasi bobot yang buruk memperparah penurunan gradien.

Mitigasi:

- Gunakan **ReLU** atau varian seperti **LeakyReLU/ELU** untuk mengurangi saturasi aktivasi.
- Terapkan **skip connections** (misalnya seperti di ResNet) untuk menghindari gradien hilang total.
- Gunakan **initialization schemes** seperti He Initialization untuk menjaga stabilitas distribusi aktivasi.

Mengapa BatchNorm Bisa Memperburuk Generalisasi:

- Jika BatchNorm diterapkan setelah lapisan konvolusi terlalu dalam tanpa fine-tuning learning rate, ia bisa menyebabkan distribusi fitur terlalu “stabil”, yang menghambat variasi penting antarkelas.
- BatchNorm juga memperkenalkan **noise dari mini-batch statistics**, yang dapat menyebabkan instabilitas saat jumlah data relatif kecil atau tidak seimbang antarkelas.

Alternatif Strategi Stabilisasi:

- **Layer Normalization** atau **Group Normalization** jika batch kecil.
- Gunakan **Dropout**, **data augmentation**, dan **early stopping**.
- Terapkan **Weight Regularization (L2)**.

2. **Ketika melatih CNN dari nol, loss training stagnan di nilai tinggi setelah XXX(3 digit epoch) epoch. Identifikasi tiga penyebab potensial terkait laju pembelajaran (learning rate), inisialisasi berat, atau kompleksitas model! Mengapa penggunaan Cyclic Learning Rate dapat membantu model keluar dari local minima, dan bagaimana momentum pada optimizer SGD memengaruhi konvergensi? Tiga Penyebab Loss Tinggi dan Stagnan:**

Tiga Penyebab Loss Tinggi dan Stagnan:

1. **Learning Rate terlalu kecil:** Update bobot sangat lambat sehingga tidak cukup signifikan untuk menurunkan loss.
2. **Inisialisasi Berat Buruk:** Jika bobot terlalu besar/kecil dari awal, aktivasi bisa masuk ke zona mati (pada ReLU) atau saturasi (pada sigmoid).
3. **Model Terlalu Kompleks:** Arsitektur terlalu dalam/lebar untuk dataset kecil, menyebabkan noise fitting atau eksplorasi ruang parameter yang tidak efisien.

Cyclic Learning Rate (CLR):

- CLR menaik-turunkan learning rate secara periodik.
- **Manfaat:** Memungkinkan keluar dari local minima atau saddle point dengan sementara menaikkan LR sehingga model bisa mengeksplorasi area parameter space lain.
- Memfasilitasi **penemuan solusi generalisasi yang lebih baik.**

Peran Momentum pada SGD:

- Momentum mempercepat konvergensi dengan **mengakumulasi gradien sebelumnya**, membantu model melintasi lembah sempit di fungsi loss dan menghindari osilasi.
- Jika dikombinasikan dengan CLR, bisa sangat efektif dalam menstabilkan sekaligus mempercepat konvergensi.

3. **Pada klasifikasi spesies ikan menggunakan CNN, penggunaan fungsi aktivasi ReLU tidak menunjukkan peningkatan akurasi setelah 50 epoch, meskipun learning rate telah dioptimasi. Jelaskan fenomena dying ReLU yang mungkin terjadi dan bagaimana hal ini mengganggu aliran gradien selama backpropagation!**

Dying ReLU:

- Terjadi ketika neuron ReLU menerima input negatif secara terus-menerus, sehingga output = 0 dan gradien = 0.
- Akibatnya, bobot neuron tersebut tidak diperbarui → **mati secara permanen.**

Dampaknya:

- Sebagian besar neuron tidak aktif, menyebabkan **hilangnya kapasitas representasi**.
- Aliran gradien terhambat → memperlambat atau menghentikan pembelajaran.

Solusi:

- Gunakan **LeakyReLU**, **ELU**, atau **PReLU** yang tetap memberikan gradien kecil saat input < 0 .
- Pastikan inisialisasi bobot tidak mendorong terlalu banyak neuron ke zona negatif.
- Kombinasikan dengan **BatchNorm** untuk menstabilkan input distribusi ke ReLU.

4. Pada pelatihan CNN untuk klasifikasi XX spesies ikan, grafik AUC-ROC menunjukkan satu kelas (Spesies X) stagnan di 0.55 sementara kelas lain mencapai >0.85 setelah YYY epoch. Analisis mengapa class-weighted loss function gagal meningkatkan kinerja Spesies X, dan identifikasi tiga faktor penyebab potensial terkait karakteristik data dan arsitektur model!

Mengapa Class-Weighted Loss Gagal:

- Class-weight hanya menambah penalti loss untuk kesalahan minoritas, tapi tidak menjamin model benar-benar belajar fitur kelas tersebut jika datanya ambigu atau tidak informatif.

Tiga Penyebab Potensial:

1. Intra-class Variability Tinggi atau Data Berkualitas Buruk:

- Citra spesies X mungkin terlalu bervariasi, buram, atau mirip dengan spesies lain.

2. Jumlah Sampel Terlalu Sedikit:

- Model tidak memiliki cukup data untuk mengenali pola dari spesies X secara konsisten.

3. Model Tidak Memiliki Resolusi/Atensi Lokal Cukup:

- Arsitektur CNN tidak dapat menangkap fitur halus yang unik pada spesies X (misal warna sirip).

Strategi Perbaikan:

- Gunakan **focal loss** daripada class-weighted loss.
- Tambahkan **attention mechanism** atau gunakan arsitektur seperti **EfficientNet** yang lebih sensitif pada fitur lokal.
- Augmentasi khusus untuk memperbanyak variasi data kelas tersebut.

- 5. Pada arsitektur CNN untuk klasifikasi ikan, peningkatan kompleksitas model justru menyebabkan penurunan akurasi validasi dari 85% ke 65%, meskipun akurasi training mencapai 98%. Jelaskan fenomena overfitting yang terjadi, dan mengapa penambahan kapasitas model tidak selalu meningkatkan generalisasi! Identifikasi 3 kesalahan desain arsitektur yang memicu degradasi performa**

Fenomena:

- Model terlalu kompleks (terlalu banyak parameter) → memorisasi data training → performa validasi menurun drastis meskipun training akurasi tinggi.

Mengapa Kapasitas Tidak Selalu Menolong:

- Kompleksitas tinggi → memodelkan noise alih-alih pola umum.
- Generalisasi buruk karena model terlalu “fleksibel” tanpa batasan eksplisit.

3 Kesalahan Desain Umum:

1. Terlalu Banyak Neuron di Fully Connected Layer:

- Menambah parameter tanpa menambah nilai fitur → overfitting.

2. Tidak Ada Regularisasi (Dropout/L2):

- Model bebas belajar apapun → kurang disiplin terhadap noise.

3. Terlalu Dalam Tanpa Residual Connections atau Normalisasi:

- Memperparah vanishing gradient dan ketidakstabilan training.